Forensic acquisition and analysis of magnetic tapes

by Bruce J. Nikkel
nikkel@digitalforensics.ch

Originally published by Elsevier in Digital Investigation
The International Journal of Digital Forensics and Incident Response
Vol. 2, No. 1 (doi:10.1016/j.diin.2005.01.007)

August 1, 2005

Abstract

Recovering evidential data from magnetic tapes in a forensically sound manner is a difficult task. There are many different tape technologies in existence today and an even greater number of archive formats used. This paper discusses the issues and challenges involved in the forensic acquisition and analysis of magnetic tapes. It identifies areas of slack space on tapes and discusses the challenges of low level acquisition of an entire length of tape. It suggests a basic methodology for determining the contents of a tape, acquiring tape files, and preparing them for forensic analysis.

Keywords: Digital Forensics, Tape Forensics, Backup Forensics, Backup tape acquisition, Magnetic tape acquisition.

Please note that "tape files" in this paper refers to SCSI tape files defined by the SCSI Stream Commands (SSC-3) standard. It does NOT refer to files contained within backup archives created by various backup programs (ARCserve, NTBackup, ufsdump, etc).

Contents

1 Introduction

This paper explains magnetic tape technology from a digital forensics perspective. It looks at tape hardware, backup/archive software, and file formats used for tape storage and highlights items of interest to forensic examiners and developers of forensic hardware. The identification of slack space on magnetic tapes is discussed, along with the issues of forensically sound tape acquisition. Some basic methods for acquiring and analyzing tape files are presented, and high level requirements for forensics capable tape drive hardware are proposed.

2 Tape hardware technologies

2.1 Historical tape technologies

Some of the earliest systems to have magnetic tape storage used large real-to-real tapes on drives connected to the system by a proprietary interface or using a standard RS232 connection. The chances of an investigator needing to analyze old real-to-real tapes are probably quite slim, but specialty tape recovery companies may exist which provide this support.

In the early 80s the SCSI interface was developed which provided a new standard for connecting devices to host systems. This started a trend towards smaller, vendor independent drives, which used small, easily removable cartridges or cassettes[1]. A popular tape format at the time was the QIC tape format. Although mostly obsolete today, if an organization has used these tapes extensively in the past, one of these drives may be a wise addition to their forensics lab.

During the 90s, the popularity of PCs with increasingly larger hard disks created a demand for inexpensive backup solutions. The industry responded with various proprietary tape technologies[2] which used floppy and IDE interfaces, parallel printer ports, or even had their own AT-bus adapter cards. This market has largely been replaced by writable CD or DVD drives and inexpensive secondary hard disks.

2.2 Common modern tape technologies

The majority of tape technologies today are found in high end workstation, server and data center environments. The most common tape technologies used over the past decade are the 4mm DDS (DAT), 8mm Exabyte,

[1] Cartridges contain a single reel of tape, while cassettes contain 2 reels.

[2] See the Linux ftape HOWTO for a good technical overview of some of these technologies

and 1/2 inch DLT. They use a SCSI interface and provide various levels of compression, density, and speed. These drives are an important part of a fully equipped corporate forensics lab. Requests to recover or analyze these tapes may arise from time to time. There are a number of good introductions and comparisons of tape technologies available which may be helpful or interesting for investigators[1][2].

There are other tape technologies on the market worth mentioning. Sony created AIT tape technology which looks like the 8mm Exabyte but is very different. IBM created Magstar tape technology which is often used in their system product line (AS/400, RS/6000, S/390). A newcomer format called LTO (Linear Tape Open) is becoming a popular standard because of its open architecture, high speed, and high capacity.

2.3 Enterprise tape libraries

When RAID systems became popular, backup technology was needed to support very large capacities. The answer was to have libraries of multiple tapes managed by robotic devices. This ranged from bolt-on tape changers for a handful of tapes, to juke-boxes with a few dozen tapes, to large robotic arms inside giant tape silos which manage libraries of hundreds or even thousands of tapes. Enterprise tape libraries have such enormous storage capacity that they are often used to perform centralized network backups of large numbers of systems.

2.4 SCSI Interface for sequential devices

Almost all professional backup systems today use the SCSI command interface[3]. The SCSI standard provides a set of relatively high level commands for accessing sequential media devices[4]. These commands allow software to specify various attributes of the tape drive such as bit densities, compression, and transfer block sizes. They allow software to read and write data as individual blocks or as entire files. Some forensic software suites may support limited analysis of tape devices through the SCSI interface, but the current SCSI standard does not allow sufficient low level access for the forensically sound acquisition of an entire length of tape.

3 Tape backup/archive software and file formats

The archive format of backup tape files is not standardized and varies depending on the backup software vendor. There are some open formats which are common among Unix and Linux systems, but most operating systems

and commercial backup solution providers use their own proprietary formats. This wide variety can create difficulties when trying to identify and recover the contents of an unknown tape.

3.1 Open backup formats

Tar stands for "tape archive". It is a very common Unix file format for creating software archives for tape or for disk storage and network transfer. Tar operates at the file and directory level. Investigators can read the tar file format with any Unix or Linux system or even with Windows based tools such as Winzip. Popular forensic software suites will also support the analysis of this common archive format.

The most common tape backup format used in Unix and Linux environments is the "dump" format. The dump command[3] creates a backup at the filesystem inode level. Sometimes larger backup systems like Amanda are based on dump and tapes can be analyzed using standard dump tools. Investigators in Unix/Linux environments will very likely need to analyze dump files on tape from time to time.

Two other traditional Unix formats for creating backup archives are pax and cpio. Though not as popular as dump and tar, an investigator may still need to recover and analyze these formats. Most Unix or Linux systems provide full support for reading these archive formats.

3.2 Proprietary backup formats

Operating system vendors often provide proprietary backup software for basic system backups. For example, Novell includes SBACKUP in Netware and Microsoft includes NTBackup with Windows NT. Some Unix vendors also include additional proprietary backup systems (HPUX and AIX for example). These formats may be difficult for investigators to analyze, especially if tapes come from high-end systems or mainframes which may not be available in the forensics lab.

There are a large number of commercial backup solutions which use proprietary tape file formats. Commercial vendors can create software with custom features, more scalability, and special hardware support for their customers. The specifications of these formats may or may not be public knowledge. Some of these formats may be supported by a lab's forensics software, if not, it may be difficult for investigators to analyze without purchasing a copy of the backup software or enlisting the support of a third party data recovery company.

[3] called ufsdump under Solaris

3.3 Features of backup software

A log or database of backup activity is often maintained by the backup software. This assists in finding the right tape if a backup has been done incrementally or is spanned across multiple tapes. This can also be useful to investigators for determining information about the existence of missing backup tapes, and possibly the existence of tapes which have been rotated out of use (retired tapes).

Most tape drives implement compression algorithms in the hardware of the tape drives (done at the tape block level). Many commercial backup software vendors also provide compression as a feature of their software. Typically one of these two compression methods (but usually not both) is used when backing up to tape. Hardware compression should not pose any difficulty to investigators since the drive can usually determine if compression was used and perform the necessary decompression. However, tape files with software compression may be difficult to analyze without the original backup software or knowledge of the compression algorithm used.

More advanced backup software systems allow for tape content to be encrypted as it is written to tape. This feature protects data when it is stored off site, possibly with a third party. Although encryption does not prevent copying or tape file acquisition, it prevents investigators from carrying out standard forensic analysis work. This is a well known problem in the digital forensics community[5][6].

Elaborate tape rotation and incremental backup systems can ensure that weekly, monthly, or even yearly backups are available for restoration. Tape rotation processes also ensure that tapes are retired once their usable lifetime has expired. This increases the complexity of an investigation, but also offers some interesting insight into the modification of files over time. Using rotated tapes of full or incremental backups can help provide a time-line of activity related to an investigation.

Tape changers, juke-boxes, and large tape silos need additional software to control the robotic retrieval and insertion of tapes. This is typically provided with the backup software and linked to a database which helps manage the tape library. Large tape libraries can be a challenge for investigators when trying to find a specific tape for analysis. An examination of the library management software and database will assist in identifying tapes of evidential interest.

4 Finding evidence at each abstraction layer

In order to have an organized understanding of tape storage details, it is helpful to separate and analyze the various layers of abstraction. At each layer, items of forensic interest can be identified. Tools and techniques needed to acquire data at each layer can also be determined.

Defined here are several layers of abstraction which could be forensically analyzed for data recovery:

- Backup/archive formatted data
- Tape files and tape partitions
- Logical blocks or records
- Tracks, frames and physical blocks
- Physical/magnetic layer

Proceeding down these layers, analysis becomes more complicated and more expensive, but also reveals more detailed information. In the following sections, a brief explanation of each layer is given followed by a discussion of opportunities and challenges for forensic analysis.

4.1 Physical/magnetic layer

At the lowest layer in the model, the magnetic fields on the physical surface of the tape could be analyzed. Magnetic force microscopy can be used to scan the tape surface. At this level of detail, it may be possible to recover certain amounts of wiped or overwritten data[7]. This is a very significant undertaking requiring advanced scanning probe microscopy equipment, but has been shown to be feasible[8].

4.2 Tracks and physical blocks

Linear and helical scan tape technologies both share the concept of physical tracks. Linear tape tracks run parallel to the edge of tape, often in a serpentine fashion (tracks can begin at either end of a tape to allow reading in both directions without rewinding). Helical scan tape tracks[4] are short strips of data written at an angle to the tape.

[4] sometimes also referred to as frames

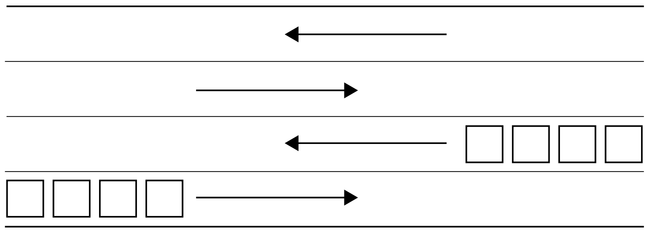

Figure 1. Physical blocks on serpentine tracks

Within these tracks, data is divided into physical blocks which typically carry additional error correction code, block headers, and other information specific to the particular tape technology. Physical blocks are usually a fixed size and may contain hardware compressed data.

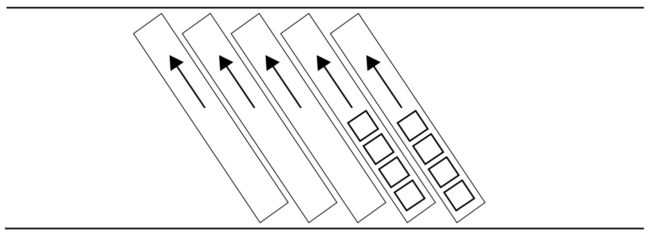

Figure 2. Physical blocks on helical scan tracks

There are no SCSI commands which allow direct access to tracks and physical blocks. Extracting data at this level requires bypassing or modifying the SCSI subsystem. A special tape drive with modified firmware would be required provide this low level access. Of possible forensic interest here are fragments of memory written to tape and found within file markers (BOP, EOP, EOD, etc.), Gap blocks, or other "housekeeping" frames. While the headers of these blocks are defined, the data portion is unused and will contain whatever additional data was in the buffer at the time of writing.

Also of interest at this layer is the tape log record. This has traditionally been recorded before the Logical Beginning Of Tape (LBOT), but newer technologies (AIT, LTO) use an EEPROM chip in the cassette or cartridge to store the information. The tape log may contain information about the tape history, error counts, the number of files and partitions, and other vendor specific information. Tape log information can be requested via the SCSI interface, usually with vendor supplied utilities. There are also third party utilities which may help[9].

4.3 Logical blocks or records

Logical blocks[5] can be variable in length and are typically comprised of one or more physical blocks (it is also possible to have multiple logical blocks within one physical block). This layer is of practical interest to digital investigators because it is the lowest layer of data that can be accessed through the SCSI interface. Software tools such as mt can be used to control blocksize, enable compression, and to position the tape at various files and logical blocks. Forensic tools (as well as the dd command) can be used to acquire data at this layer, although not to the same extent as with hard disk forensics.

There is one significant problem with doing tape acquisitions at this layer. The SCSI interface specification does not provide commands to access logical blocks beyond the End-Of-Data marker (located immediately after the last file on the tape). Any data from previous backups which resides beyond the EOD cannot be recovered using standard software tools.

4.4 Tape files and partitions

Tapes do not have a hierarchical filesystem structure like regular filesystems. They store files sequentially on a tape or within a tape partition[6]. Partitioning of a tape (sometimes referred to as tape "directories") can allow the grouping files. This makes it possible to erase and overwrite sections in the middle of the tape. Files are accessed by moving or "spacing" forward or backward to the beginning of a file number, and then proceeding to read logical blocks until the end of the file.

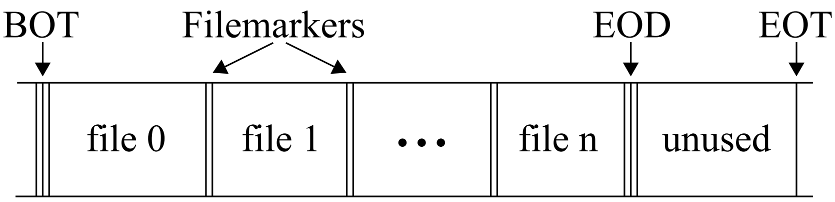

Figure 3. Files and markers on a tape

This level of simplicity makes it easy for investigators to move tape data onto a harddisk for analysis, or to copy it to a new tape for restoration. Tape files can be transferred to an investigation machine for analysis using the dd

[5]sometimes referred to as records
[6]Sometimes called a flat filesystem

command or possibly by acquisition utilities provided by forensic software. All backup software, including proprietary programs, use this simple tape file standard to store archive data on tape.

4.5 Backup/archive formatted data

Some backup software, like dump or tar, will store the archive data in a single tape file. Other backup software, like NTBackup, may use several tape files to store archive data along with other meta data. Additional files can be appended to the end of a tape (additional backups, incremental backups, etc.). When copying the files for forensic purposes, it is important to copy every file on the tape (up to the EOD marker).

Incremental backup files add a certain element of complexity to an investigation, but they also have some advantages that even harddisk forensics cannot offer. Having incremental backups of changes over a period of time gives an investigator some insight into filesystem activity over time. By analyzing the changes of files in daily incremental backups, a time-line of activity can be built which may assist in solving an case.

5 Tape slack space and unused blocks

5.1 Comparison to hard disk slack space

The notion of slack space in digital forensics originally comes from hard disk analysis and describes the unused space at the end of a file. Sectors in an allocated cluster or block which are unused by the file are called file slack and may contain data from previously deleted or overwritten files. If a single sector is partially used by the file, the unused portion will contain whatever was in the memory buffer at the time of writing. This area is called memory slack, buffer slack, or RAM slack.

Tape file blocks are not allocated with pointers as files on a hard disk filesystem. Tape files are written sequentially starting from the beginning of a tape or partition, and each tape file is a contiguous sequence of blocks. Individual physical blocks are not re-allocated to new files as with hard disk files. For this reason, the concept of file slack does not apply to tapes. However, tape files may exhibit RAM slack at the ends of files and within the various file markers and GAPs. These areas of RAM slack are discussed here[7].

[7] Some of these may not apply to certain tape technologies

The physical blocks on some tapes look almost like network packets, they contain a header and a data portion. In some cases, the data portion of a physical block may be undefined and will contain whatever data was in the memory buffer at the time of writing. Detailed information about the different tape block layouts can be found at the ECMA website (http://www.ecma-international.org/). This site contains useful information about the physical standards of various tape formats such as DDS[10], DLT[11], 8mm[12], and others.

5.2 Some definitions

Tape file RAM slack represents the data found between the logical end of a tape file and the end of the last logical tape block. The backup software or operating system may have padded this area with an arbitrary chunk of memory before the logical block was written to tape.

If a logical block size is variably defined, the end of a given logical block may reside within a physical block. The area from the end of the logical block to the end of the physical block (where it resides) may be undefined and contain arbitrary data from the tape drive's memory buffer. This area will be referred to as *block RAM slack*.

The various file markers on a tape often consist of a number of contiguous physical blocks with undefined data portions. The data found in these undefined areas of a file marker will contain arbitrary data from the tape drive's memory buffer. This area will be referred to as *file marker RAM slack*.

Different tape technologies may use gap blocks to pad various areas of tape (for example, some helical scan tape drives will pad an incomplete track with gap blocks when finishing a write operation). The data portion of these Gap Blocks is often undefined and may contain arbitrary data from the tape drive's memory buffer. These areas will be referred to as *gap RAM slack*.

Physical blocks which exist beyond the EOD marker of a tape or tape partition may contain data from previous backups. These tape blocks shall be referred to as *unused tape blocks* (not slack space). This includes tapes which are "short erased", where a new EOD is simply written at the beginning of the tape. Tapes which are "long erased" typically write erase blocks with an used data portion across the entire tape. This unused data area will contain whatever was in the tape drive's memory buffer at the time of writing.

5.3 Usefulness of tape slack areas

Since tape slack areas generally do not contain data from previous backups, they are arguably less useful for forensics practitioners. It is mentioned here mainly for completeness and is an area where more research could be done. One potential use for these slack areas could be in the analysis of encrypted tapes. It is possible that certain slack space areas on tapes could contain small amounts of clear text information which could be of use to an investigator.

6 Preparing for tape acquisition

6.1 Adherence to existing acquisition standards

The forensically sound acquisition of direct access media such a hard disks, floppies, etc. has been well documented in the past[13][14][15]. It is important to build on this work where possible and follow existing standards of evidence collection. NIST[16] has outlined some requirements for disk imaging tools. These requirements can be adapted to the imaging of tapes as follows:

- the tool shall make a bit-stream duplicate image of an original tape
- the tool shall not alter the original tape
- the tool shall be able to verify the integrity of a tape image
- the tool shall log I/O errors
- the tool shall be properly documented

The obvious problem is the lack of low level tape access which prevents creation of a bit-stream duplicate. Verifying the integrity of a tape could also be an issue since it requires multiple read passes of a tape, something which is may be undesirable under certain conditions.

6.2 Physical handling of tapes

Tapes are not well sealed in a protective enclosure like harddisks. They are sensitive to environmental factors such as humidity, dust, and smoke, etc. This makes the tapes vulnerable to handling and even reading. Physical storage of tape evidence is also critical.

Since every physical read of a tape (especially old or damaged tapes) could cause damage to the evidence, attempts should be made to acquire data with as few read passes as possible. If possible, the entire acquisition should take place with a single read pass.

6.3 Write blocking

To ensure that an acquisition procedure is forensically sound, steps must be taken to ensure that the original data is not modified or altered in any way. Hard disk forensics has solved this problem through the use of write blocking devices[17][15]. Tape drives make this process easier because each tape has a physical read-only tab or switch which should prevent modification[8].

6.4 Levels of acquisition

It has already been mentioned that the SCSI command set does not allow low level access to physical blocks. This hampers the ability to create the preferred bit-stream image. However, using standard hardware and software tools, regular tape file data can still be recovered and used as evidence. In many cases this may be sufficient for an investigation.

The last two sections of this paper both cover the acquisition of tapes, but they take different approaches, and have different requirements. The first section deals with acquiring a low-level image of an entire length of tape. This is the ideal scenario, but requires specialized hardware or modified tape firmware to complete. The second section demonstrates a more practical approach which simply copies every file from the tape for analysis. Although the slack space and post-EOD data blocks cannot be recovered, the tape files may still be of evidential value.

6.5 Cryptographic hash of tape

When extracting the bit-stream image of an entire length of tape, a single cryptographic hash is enough to preserve evidence integrity of the media. For a file level acquisition of a tape, a cryptographic hash of each individual file must suffice.

7 Acquiring tape blocks across the entire length of tape

7.1 The need for low level tape imaging

In order to conform to accepted standards for complete and forensically sound acquisition, low level imaging of an entire length of tape is needed. Current methods of acquiring and analyzing tapes are not at the same level of completeness that is common among other forms of storage media. The

[8]in the past, this engaged a physical write-protection to protect the media. Today, the mechanism may be implemented in firmware and could theoretically be overridden.

use of tapes will be around for a number of years to come and forensic tools to analyze them in a more complete manner need to be created.

7.2 Problem of low level access

To acquire a tape in a forensically complete manner, it is desirable to access every physical block or byte on the tape. This would allow access to data stored in gap blocks, the various file and partition markers, as well as data between the end of data (EOD) and the physical end of tape (PEOT). This is problematic since the SCSI interface abstracts the user from this level of access. The SCSI Stream commands[4] allow reading of existing tape files or their corresponding logical blocks. Reading physical blocks of data at arbitrary points on a tape (especially within filemarks or beyond the EOD) is not supported. This means that general software for analyzing tape cannot be created and a firmware/hardware solution is required.

7.3 Attempted solutions to the EOD problem

There have been a number of (mostly informal) suggestions in the past for reading past the EOD file marker and acquiring data over the full length of tape. These methods have several problems when used for forensic purposes.

One method suggests writing a small file just before the EOD, but powering off the drive before it has a chance to finish (thus overwriting the EOD). This method has several obvious problems making it unsuitable for forensic use. It modifies the data on the tape, breaking the cardinal rule of evidence preservation. It is also not reproducible and therefore not verifiable or reliable. There is no known error rate. There is a high risk of overwriting and losing evidence.

Another method suggests finding the approximate location of the EOD and physically splicing the tape to remove it. The problems here are the same as the previous example with the added risk of causing other physical damage to the tape.

Another idea suggests spacing up to the EOD and then physically advancing the tape past the marker. This introduces the risk of damaging both the tape and the drive. The degree of error and reproducibility is also questionable.

Yet another (slightly better) method relies on changing drive firmware parameters[9] to allow accessing data beyond the EOD marker. This exploits

[9]usually using a diagnostic serial cable attached to the drive electronics

firmware features to trick the drive head into moving past the EOD marker. While this may protect the tape from modification, the method is very specific to vendor and to individual models of tape drive.

A more difficult method is to create customized or modified firmware for a particular drive which explicitly allows low level access to an entire length of tape. This is a significant undertaking and requires either reverse engineering the firmware or possibly licensing the sourcecode from a particular vendor. This method is sometimes used by professional data recovery services but may be too expensive and time consuming to build for a typical forensics lab.

A low level interface to tape drive devices was proposed as a SCSI standard in 1986, but withdrawn in 1997[18]. The standard described access to the hardware interface, bus timing commands, and status. This standard may have had some potential for forensic use.

Most of these methods break a number of accepted practices regarding the collection of evidence. There are issues of modifying or destroying evidence, questionable reliability and error rates, and lack of general acceptance within the digital forensics community. All of these issues could play a role in the admissibility of tape evidence. A solution which solves these problems is needed.

7.4 Solution Requirements

This low level access problem creates a need for specially modified, forensics capable tape drive devices. Outlined here are some basic requirements for such a solution.

The reading of physical tape blocks is sufficient to provide a forensically sound acquisition (the same level of quality as hard disk sectors). The data recovered should include the various tape markers and gaps as well as data beyond the EOD. The acquired data should represent a sequence of physical tape blocks from the PBOT (Physical Beginning Of Tape) to the PEOT (Physical End Of Tape).

It should create an acquired file in a vendor-independent format that can be imported into standard forensics software suites for analysis. Just as a dd disk image typically represents a linear sequence of 512 byte sectors, a tape image should contain a linear sequence of physical blocks (abstracting hardware particulars such as serpentine tape layouts, linear and helical scan heads, etc.)

The interface for acquiring an entire length of tape should be hardware independent as much as possible (ie. should work with DDS, DLT, 8mm, LTO, etc. including future tape technologies). Adding a forensic acquisition capability to the existing SCSI command set would be ideal.

It should be a read-only function. Tapes do have a Read-Only tab, but ensuring the method is implemented in a read-only manner reduces potential human error.

It should be accessible as a separate raw device which could be read with forensic tools or dd in a standardized manner.

It should recognize and report errors but continue reading a tape. This is especially crucial for reading past sections of tape which may be physically damaged.

If possible, it should be implementable on existing tape drives with a firmware upgrade.

8 Acquiring and analyzing files from a tape

The previous section identifies the difficulties of acquiring an entire length of tape and shows that byte level access to tapes using standard software tools is not feasible. However, in many cases, just recovering the normal files on a tape is enough to help an investigation. This section outlines a practical method for acquiring and analyzing these files.

8.1 Forensic acquisition tools

There are feature rich commercial tools on the market (for example, Tapecat and MMPC) which acquire and analyze tape files and have vendor support. For the following examples, however, we use a Linux system. A typical Linux installation comes with an enormous set of powerful, free tools, including those needed for the extraction and analysis of tape file data.

The mt (magnetic tape) command is used to control a SCSI tape from the command line. It allows moving a tape forward or backward, changing parameters, erasing, and ejecting tapes.

The dd command can be used to copy blocks from tape to disk, or from disk to tape. Although usage here is similar to methods used in hard disk acquisitions, dd behaves differently when operating on tape devices.

Individual tape drive vendors may provide additional tools for diagnostic purposes. These tools may be useful for extracting vendor specific information from the tape log.

8.2 Preparation

The preparation for this investigative session is the same as that outlined in [19]. It consists of creating an evidence directory, starting the script command to log the session and ensuring the correct time and date is recorded before proceeding with the investigative work[10].

Before we start, we need to identify the operating system's *non-rewinding* tape device. Working with this device will ensure that the tape is not automatically rewound each time we perform a tape operation. Under Linux this is usually /dev/nst0, Solaris is normally /dev/rmt/0n and FreeBSD typically uses /dev/nsa0. The non-rewinding device will allow us to copy all files from the tape in a single read pass.

Each tape vendor specifies the typical compressed capacity of their drive. Make sure there is enough disk space to acquire a complete uncompressed copy of a tape.

Be sure the tape's read-only tab is engaged and insert the tape. Once inserted, we can view the tape status with the mt command as follows.

```
# mt -f /dev/nst0 status
```

This will show status information about the drive and tape, including the position of the tape, and possibly a read-only indicator.

8.3 Acquisition

The dd command will copy logical tape blocks until the end of a tape file. To recover every file on a tape, we must repeatedly issue dd commands until we receive an error indicating that we are at the end of the written tape.

```
# dd if=/dev/nst0 of=file0.dd bs=128k
# dd if=/dev/nst0 of=file1.dd bs=128k
# dd if=/dev/nst0 of=file2.dd bs=128k
# dd if=/dev/nst0 of=file3.dd bs=128k
# dd if=/dev/nst0 of=file4.dd bs=128k
...
```

[10]This procedure may require root access on some systems

Note how the input file remains the same for each command. Because we are using a non-rewinding tape device, every dd copy operation will position the tape head at the next file, ready for the next command.

The last two files created may be zero length. These come from the EOD marker and can be ignored. The mt status command can be used to verify that the tape is positioned at the end of the data.

When reading tape files with dd, the specified block size must be greater than or equal to the block size used to originally write the file. Since we don't know this in advance, we guess. A block size of 128k should work in most cases, if not, an error will be generated and the block size will need to be increased. The output from these dd commands should be saved. If we need to create duplicate tapes, we can use this output to calculate the correct block size for writing (this will be demonstrated later).

Once all the files have been recovered, we can remove the tape and take steps to preserve the evidence.

8.4 Evidence Preservation

The physical preservation of the tape is outside the scope of this paper, but proper storage conditions such as temperature and humidity are available from the tape vendor (typically these specifications are included in the sleeve of the tape case).

The cryptographic hashes[11] of each tape file should be created, printed, and stored together with the tape. The method used is the same as hashing dd images of hard disks. Here we take the hash of all dd tape files in our evidence directory:

```
# md5sum *.dd > hash.txt
```

At this point, we have acquired the tape data and taken steps to preserve the evidence. We now move on to analyzing the recovered data.

8.5 Analysis of tape files

We now have the tape's contents stored as ordinary files on our local hard disk. Our first task is to determine the software used to create those files. In some cases, such as dump and tar, the unix file command can be used to determine the backup type:

[11] This command may vary depending on the system. The md5 algorithm may also be replaced with SHA-1 if desired

```
# file *.dd
```

If there is a recognized Unix file, its type will be shown. If the files are not recognized, then we will need to manually look at them. Standard forensic analysis methods and techniques, such as using the strings command or a hex editor, can be used to determine the file type. In this example, we have extracted an NTBackup archive[20]. This was quickly determined by examining the first file, which happened to contain the following strings:

```
Media created 29.12.2004 at 08:17MTF Media Label|1.0|Seagate|NTBackup5.0|

Microsoft Windows NT Backup (NTBACKUP.EXE) Version 1.0 Rev. 3.41
```

Backup files from other proprietary vendors may also contain useful identification strings which could aid in determining the software used to create the archive.

8.6 Restoring content using the original backup software

In some cases, forensic software analysis tools may be able to operate directly on the extracted data (tar for example). If not, the tools used to create the archive may be able to restore archive data from the files (dump for example). There may also be special conversion or extraction tools available (like vmsbackup, a unix tool for reading vms tapes).

If it is infeasible to extract data directly from the recovered tape files, it may be necessary to create a duplicate tape and perform a tape restore using the original backup software. This is done by copying the acquired files back to a new tape using dd in a manner similar to the file acquisition method.

Before doing this, it is important to know the correct block size for each file. Writing archive files to tape with the wrong block size may cause problems with some backup software (NTBackup for example). The original block size can be calculated from the dd output received during file acquisition. For example:

```
0+2985 records in
0+2985 records out
195624960 bytes transferred in 66.968103 seconds (2921166 bytes/sec)
```

The block size is calculated by dividing the number of bytes transferred by the number of records. In this example, 195624960 divided by 2985 gives us a block size of 65536, which can be used to write the file to tape.

Having calculated the block sizes for each file, we insert a new tape of equal or larger size and transfer the files using dd.

```
# dd if=file0.dd of=/dev/nst0 bs=65536
# dd if=file1.dd of=/dev/nst0 bs=65536
# dd if=file2.dd of=/dev/nst0 bs=65536
...
```

Once completed, the tape can be restored in a normal fashion, using the original software determined in the previous section. Be careful to restore the backup tape into its own evidence directory. Once restored, standard forensic analysis (searching, hashsets, etc.) can then be used on the restored data.

In cases where the original backup software is no longer available, or obsolete, data may still need to be recovered. Many data recovery companies exist which provide these services.

8.7 Analysis of other tape artifacts

There may be other items of interest to an investigator when analyzing a case involving backup tapes.

As previously mentioned, each tape may contain a log area at the beginning of the tape or on an EEPROM chip. Vendors may store additional information here. Check with the device vendor or third parties for diagnostic tools which may be used to extract this data.

If the system used to create the tape is available for analysis, it may be possible to discover information about the existence of other tapes. The backup software may have kept a record of backups made in the past For example, the dump command may update the *dumpdates* file with every backup. NTBackup also keeps a log of backup activity.

The contents of the tape may reveal a partial backup which is part of a multi-tape, spanned backup. It may also be an incremental backup, indicating the existence of additional tapes.

9 Concluding remarks and future work

Very little has been published on the forensically sound acquisition and analysis of tapes. It is hoped that this paper will provide a useful starting point for more tool development in this area. Tapes continue to be used in many environments, and old tapes continue to be recovered for analysis.

Here are some examples of additional practical work that could be done in the area of backup tape forensics. These items could very useful to the digital forensics community:

- Develop special forensic tape drives to read tapes at the byte or physical block level from beginning to end of the physical tape.
- Create and maintain a database of backup software file signatures to assist in identifying the contents of backup archives. This need not be limited to tape technologies, but could be useful for any backup medium.
- Submit a request to the SCSI standards committee for block level reads of an entire length of tape for forensic purposes.

References

[1] Catherine DeGraff. Tape Drive Technology Comparison. Spectra Logic, Nov 2001

[2] StorNet Solutions. Tape Drive Technologies Primer. http://www.storenetsolutions.com, downloaded Dec 2004

[3] SCSI Primary Commands - 3 (SPC-3). Project T10/1416-D Working Draft, Dec 2004.

[4] SCSI Stream Commands -3 (SSC-3). Project T10/1611-D Working Draft, Aug 2004.

[5] Eoghan Casey. Practical Approaches to Recovering Encrypted Digital Evidence. International Journal of Digital Evidence, Fall, 2002

[6] Jason Siegfried, Christine Siedsma, Bobbie-Jo Countryman and Chester D. Hosmer. Examining the Encryption Threat. International Journal of Digital Evidence, Winter 2004

[7] R. D. Gomez, A. A. Adly, I. D. Mayergoyz. Magnetic Force Scanning Tunneling Microscope Imaging of Overwritten Data. IEEE Transactions on Magnetics, Vol 28, No.5, 1992

[8] Peter Gutmann. Secure Deletion of Data from Magnetic and Solid-State Memory. 6th Usenix Security Symposium, 1996

[9] TapeRx Users Guide Version 4.5. Certance LLC, 2004

[10] ECMA. 3.81mm Wide Magnetic Tape Cartridge for Information Interchange - Helical Scan Recording - DDS-3 Format using 125m Length Tapes, 1996

[11] ECMA. Data Interchange on 12.7mm 128-Track Magnetic Tape Cartridges - DLT 4 Format, Dec 1995

[12] ECMA. 8mm Wide Magnetic Tape Cartridge for Information Interchange - Helical Scan Recording - DA-2 Format. Dec 1996

[13] Anthony F. DeSante. Evidentiary Considerations for Collecting and Examining Hard-Drive Media. George Washington U, 2001

[14] US Dept. of Justice (CCIPS). Searching and Seizing Computers and Obtaining Electronic Evidence in Criminal Investigations, 2002

[15] NIST Computer Forensics Tool Testing (CFTT) Project. Software Write Block Tool Specification and Test Plan, 2003

[16] NIST Computer Forensics Tool Testing (CFTT) Project. Disk Imaging Tool Specification, 2003

[17] NIST Computer Forensics Tool Testing (CFTT) Project. Hardware Write Blocker Device (HWB) Specification, 2004

[18] Press Release of SCSI standard withdrawal: http://www.incits.org/press/1997/pr97008.htm

[19] B. J. Nikkel. Domain name forensics: a systematic approach to investigating an internet presence. Digital Investigation Vol 1 No 4, 2004

[20] Seagate Software. Microsoft Tape Format Specification, Version 1.00a, Seagate Software, Inc. 2000

Acknowledgments: I would like to thank Paul Sanderson from Sanderson Forensics for his advice in helping me correct some errors which were in the originally published version of this paper.

Document History

Dec 2004 - Jan 2005: Created original article
Jan 26, 2005: Submitted to Elsevier
March 14, 2005: Published in Digital Investigation Journal
April 23,2005: Corrected slack space errors. Reworded several paragraphs. Added a paragraph on the usefulness of slackspace. Added a word of thanks to Paul Sanderson.
April 24,2005: Sent letter of correction to Elsevier. Made a corrected version of the paper freely available on the Internet.
July 6,2005 Added a clear, boldface note on the title page to ensure no misunderstanding of the terms 'tape files'.
August 1,2005 Added table of contents and document history